Research Paper | Computer Science | E-ISSN No : 2454-9916 | Volume : 10 | Issue : 9 | September 2024

GAUSS JACOBI METHOD IMPLEMENTED IN PYTHON PROGRAMMING LANGUAGE

M. Suganya Veni[1], S. Devi Durga[2], S.A. Sowmiya[3]

[1] M.Sc., MPhil in Mathematics, Assistant Professor, Madurai Institute of Social Sciences, Madurai, India
[2] M. Sc Computer Science, Assistant Professor, Madurai Institute of Social Sciences, Madurai, India
[3] M. Sc. Computer Science, Assistant Professor, Madurai Institute of Social Sciences, Madurai, India

ABSTRACT

Numerical method plays a vital role in solving real time problems that are related to Mathematics, Science, Business, Medicine and Engineering etc. As we all know that every technology is related to mathematics. Hence, among all methods present in Numerical methods, through this paper we are going to discuss about Gauss Jacobi method by considering a problem, with the parallel solution is implemented in trending programming language as in PYTHON

KEYWORDS: Gauss Jacobi method, Python, Iteration, Converge, Approximate

INTRODUCTION

The evolution of numerical methods on a daily basis is to discover the right solution techniques for solving problems in the field of applied science as well as pure science, for example: weather forecasting, population analysis, studying the spread of a disease, predicting chemical reactions, physics, optics, etc.

Numerous practical mathematics issues necessitate solving linear equation systems, where the linear system may arise naturally in some circumstances and, in other situations, as a step in the resolution process. The linear equation collections are referred to as linear equation systems. They include the same group of variables.

Systems of linear equations can be solved using a variety of techniques, although no single technique is ideal in every circumstance. These techniques ought to be chosen based on accuracy and quickness. Furthermore, because there are a lot of calculations required to solve big systems of equations, speed is essential.

The majority of linear equation systems are found to arise in many different contexts, both directly and indirectly through the numerical solutions of other mathematical models and in the modelling of actual circumstances. These applications are evident in almost every field of study, including biology, social science, and physics.

A linear equation in the variable
Y1,Y2,Y3…… Yn is any equation of the form a1y1+a2y2 +...+anyn=b1.

Systems of linear equations can be solved in a variety of ways, including direct and indirect (iterative) methods. When there is no error other than the rounding off error brought on by machine processes, the direct techniques always give us the exact solution, whereas the iterative methods only offer us the approximate solutions.

There are two types of approaches for solving linear systems of equations: 1) direct approaches and 2) iterative methods. When there are many equations in a system, especially when the coefficient matrix is sparse, direct approaches are not suitable. The Gauss Elimination approach [1] presented an issue for the example due to round-off errors and sluggish convergence for large systems of equations. Iterative techniques are very efficient in terms of time and computer storage needs.

GAUSS JACOBI

The iterative method known as the Jacobian Method (sometimes called the Jacobi Method) is used to solve systems of linear equations. It is especially helpful in large systems where computationally expensive direct approaches (like Gaussian elimination) [2] are used. German mathematician Carl Gustav Jacob Jacobi is credited with developing the method. This method is used to solve strictly diagonally dominant systems of linear equations [6]. It involves solving each diagonal element for an approximate value, and then repeating the process until it converges.

The everyday development of numerical methods is to identify appropriate problem-solving strategies for both pure and practical science, such as weather. It is anticipated that the system of linear equations can be represented by the following expression: The formula is Axe = b, where x is the vector of unknown variables, b is the vector of constants, and A is the coefficient matrix. The Jacobi method proceeds as follows:

- Assumption 1: The system of linear equations has a unique solution.
- Assumption 2: The coefficient matrix A has no zeros on its main diagonal.

If any of the diagonal entries are zero, then rows or columns must be interchanged to avail a coefficient matrix that has non-zero entries on the main diagonal.

PYTHON PROGRAMMING LANGUAGE

One well-liked programming language is Python. Guido van Rossum was the creator, and it was published in 1991. It is utilised in system scripting, mathematics, software development, and server-side web development. This programming language has attributes such as object-oriented, high-level programming, and interpretation. Because of its easy-to-learn syntax, it became the obvious choice for people just starting out in programming. Making it simpler to read and comprehend for developers while also cutting down on the number of lines of code was the main goal in its creation.

GAUSS JACOBI METHOD

Let us explain this method is the case of linear equations in three unknowns.

Consider the system of linear equations,

$$a_1x+b_1y+c_1z=d_1$$

$$a_2x+b_2y+c_2z=d_2$$

$$a_3x+b_3y+c_3z=d_3$$

For Converges

$$|a_1|>|b_1|+|c_1|$$

$$|b_2|>|a_2|+|c_2|$$

$$|c_3|>|a_3|+|b_3|$$

For iterative process

$$X= \frac{1}{a_1}(d_1 - b_1 y - c_1 z)$$

$$y = \frac{1}{b_2}(d_2 - a_2 x - c_2 z)$$

$$z = \frac{1}{c_3}(d_3 - a_3 x - b_3 z)$$

If $x^{(0)}$, $y^{(0)}$, $z^{(0)}$ are the initial values of x, y, z respectively, then

$$x^{(1)} = \frac{1}{a_1}(d_1 - b_1 y^{(0)} - c_1 z^{(0)})$$

$$y^{(1)} = \frac{1}{b_2}(d_2 - a_2 x^{(0)} - c_2 z^{(0)})$$

$$z^{(1)} = \frac{1}{c_3}(d_3 - a_3 x^{(0)} - b_3 z^{(0)})$$

Again using these values $x^{(1)}$, $y^{(1)}$, $z^{(1)}$ we get

$$x^{(2)} = \frac{1}{a_1}(d_1 - b_1 y^{(1)} - c_1 z^{(1)})$$

$$y^{(2)} = \frac{1}{b_2}(d_2 - a_2 x^{(1)} - c_2 z^{(1)})$$

$$z^{(2)} = \frac{1}{c_3}(d_3 - a_3 x^{(1)} - b_3 z^{(1)})$$

Proceeding in the same way , if the rth iterates are $x^{(r)}$, $y^{(r)}$, $z^{(r)}$ the iteration scheme reduces to

$$x^{(r+1)} = \frac{1}{a_1}(d_1 - b_1 y^{(r)} - c_1 z^{(r)})$$

$$y^{(r+1)} = \frac{1}{b_2}(d_2 - a_2 x^{(r)} - c_2 z^{(r)})$$

$$z^{(r+1)} = \frac{1}{c_3}(d_3 - a_3 x^{(r)} - b_3 z^{(r)})$$

The procedure is continued till the convergence is assured (correct to required decimals).

Problem:

Solve the following equation by Gauss Jacobi Method.

$$3x + 4y + 15z = 54.8;$$

$$x + 12y + 3z = 39.66;$$

$$10x + y - z = 7.74.$$

Solution:

n	X	Y	Z
1	0.774	3.305	3.6533
2	1.1742	2.3272	2.6172
3	1.0647	2.5529	2.7979
4	1.0783	2.5168	2.7596
5	1.0742	2.5252	2.7665
6	1.0748	2.5239	2.7651
7	1.0746	2.5242	2.7653
8	1.0747	2.5241	2.7653

IMPLEMENTATION USING PYTHON LANGUAGE:

The above problem is implemented in python programming.

Coding:

```
jacobi.py - C:\Users\devid\AppData\Local\Programs\Python\Python39\jacobi.py (3.9.13)
File  Edit  Format  Run  Options  Window  Help
f1 = lambda x,y,z: (7.74 -y +2*z)/10
f2 = lambda x,y,z: (39.66 -x -3*z)/12
f3 = lambda x,y,z: (54.8 -3*x -4*y)/15

# Initial setup
x0 = 0
y0 = 0
z0 = 0
count = 1

# Reading tolerable error
e = float(input('Enter tolerable error: '))

# Implementation of Jacobi Iteration
print('\nCount\tx\ty\tz\n')

condition = True

while condition:
    x1 = f1(x0,y0,z0)
    y1 = f2(x0,y0,z0)
    z1 = f3(x0,y0,z0)
    print('%d\t%0.4f\t%0.4f\t%0.4f\n' %(count, x1,y1,z1))
    e1 = abs(x0-x1);
    e2 = abs(y0-y1);
    e3 = abs(z0-z1);

    count += 1
    x0 = x1
    y0 = y1
    z0 = z1

    condition = e1>e and e2>e and e3>e

print('\nSolution: x=%0.3f, y=%0.3f and z = %0.3f\n'% (x1,y1,z1))
```

Output:

```
IDLE Shell 3.9.13
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\devid\AppData\Local\Programs\Python\Python39\jacobi.py ===
Enter tolerable error: 0.0001

Count   x       y       z

1       0.7740  3.3050  3.6533

2       1.1742  2.3272  2.6172

3       1.0647  2.5529  2.7979

4       1.0783  2.5168  2.7596

5       1.0742  2.5252  2.7665

6       1.0748  2.5238  2.7651

7       1.0746  2.5242  2.7654

8       1.0747  2.5241  2.7653


Solution: x=1.075, y=2.524 and z = 2.765
```

The Actual Values are x=1.075, y= 2.524 and z= 2.765

CONCLUSION

There are different methods of solving of linear equation some are direct methods while some are iterative methods. In this paper, Gauss Jacobi Iteration method of solving of linear equation have been presented and implemented in PYTHON through the IDLE software for the better understanding to all the Mathematics and computer science students.

REFERENCES

1. " Jacobi and Gauss-Seidel Iterative Methods for the Solution of Systems of Linear Equations Comparison" Vishal V. Mehtre, Aditya Singh Internal Journal for Information Research in Technology, December 2019
2. "Comparision of Gauss Jacobi Method and Gauss Seidal Method using Scilab ", S. Sathya, T.Ramesh International Journal of Trend in Scientific Research and Development (IJTSRD) October 2019
3. Doha, E. H., et al. "A new Jacobi rational–Gauss collocation method for numerical solution of generalized pantograph equations." Applied Numerical Mathematics (2014).
4. Varona, Juan L. "Graphic and numerical comparison between iterative methods." Mathematical Intelligencer (2002).
5. Ahmadi, Afshin, et al. "A parallel Jacobi-embedded Gauss-Seidel method." IEEE Transactions on Parallel and Distributed Systems (2021).
6. Wang, Tao, et al. "Implementation of Jacobi iterative method on graphics processor unit." 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems.